Package: ecoXCorr (via r-universe)

February 5, 2026

Type Package

Title Lagged Cross-Correlation Analysis of Environmental Time Series

Version 0.1.0

Description Tools for analysing lagged relationships between environmental variables and ecological or epidemiological time series. The package implements a workflow to aggregate meteorological data over multiple lagged intervals, fit regression models for each lag window, and visualise effect strength and direction using cross-correlation maps (CCM).

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.3

Imports glmmTMB, ggplot2, stats, performance

Depends R (>= 3.5)

URL https://github.com/Nmoiroux/ecoXCorr

BugReports https://github.com/Nmoiroux/ecoXCorr/issues

Config/pak/sysreqs cmake make libicu-dev libssl-dev

Repository https://r-multiverse-staging.r-universe.dev

Date/Publication 2026-02-04 14:27:32 UTC

RemoteUrl https://github.com/Nmoiroux/ecoXCorr

RemoteRef v0.1.0

RemoteSha f12e78a6c969e4300ef7646f28c6e75ae6fc7548

Contents

aggregate_lagged_intervals 2
albopictusMPL2023 3
fit_models_by_lag 4
meteoMPL2023 6
plotCCM 7

Index 9

`aggregate_lagged_intervals` *Aggregate meteorological time series over multiple lagged time intervals*

Description

This function computes aggregated values of one or several meteorological time series over all possible lagged time intervals defined relative to one or more reference dates. For each reference date `d`, all intervals $[d - k \times i,\ d - (l-1) \times i)$ are generated, where `i` is the interval length (in days) and `k, l` range from 1 to `m` with `k >= l`. Each interval is then used to aggregate the specified meteorological variables using one or more summary functions.

Usage

```
aggregate_lagged_intervals(
  data,
  date_col,
  value_cols,
  d,
  i = 1,
  m,
  funs = list(mean = mean, min = min, max = max, sum = sum),
  na.rm = TRUE
)
```

Arguments

`data`	A data.frame containing the meteorological time series.
`date_col`	Character string giving the name of the date column in `data`. The column must be of class `Date` or `POSIXct`.
`value_cols`	Character vector giving the names of numeric variables to be aggregated (e.g. rainfall, temperature).
`d`	Vector of reference *dates*. Can be of class `Date` or coercible to `Date`. Aggregations are computed independently for each date.
`i`	Integer giving the length of the base time *interval*, expressed in days (e.g. `1` for daily data, `7` for weekly intervals, `14` for fortnightly intervals).
`m`	Integer giving the *maximum* lag (number of intervals) to consider. All combinations of lag windows with `1 <= lag_end <= lag_start <= m` are evaluated.
`funs`	Named list of aggregation functions to apply to each variable. Each function must accept a numeric vector as first argument. The names of the list are used to construct output column names (e.g. `rain_mean`, `temp_max`). Defaults to mean, min, max and sum.
`na.rm`	Logical indicating whether missing values should be removed before aggregation. Passed to the aggregation functions (default: `TRUE`).

Details

The function supports multiple reference dates, multiple variables, and multiple aggregation functions, and returns all combinations as additional columns in the output data frame.

Reference dates for which at least one interval contains no observations are reported in the console as having missing data. Reference dates for which at least one interval partially lies outside the temporal bounds of the input time series are reported as having truncated intervals.

Intervals are defined as left-closed and right-open (`[start, end)`). An interval is considered truncated if it extends beyond the temporal bounds of the input time series. An interval is considered missing if no observations fall within it.

Console messages are printed to inform the user of reference dates for which missing data or truncated intervals occurred.

Value

A data.frame with one row per reference date and lag window, containing:

- `date`: reference date
- `start`, `end`: start (inclusive) and end (exclusive) of the aggregation interval
- `lag_start`, `lag_end`: lag indices defining the interval
- One column per combination of variable and aggregation function (e.g. `rain_mean`, `temperature_sum`)

`albopictusMPL2023` *Entomological records of female *Aedes albopictus* in Montpellier (2023)*

Description

This dataset contains entomological sampling records of female *Aedes albopictus* collected in Montpellier (France) during 2023. The data originate from GBIF and correspond to a zero-truncated subset (i.e. only positive captures) of adult female mosquitoes collected using fixed traps across different areas of the city.

Usage

```
albopictusMPL2023
```

Format

A data.frame with the following variables:

species Scientific name of the species (*Aedes albopictus*).

individualCount Number of female individuals captured during the sampling event.

eventDate Original event date as provided by GBIF.

trap Identifier of the trap used for sampling.

area Identifier of the area where the trap is located.

date Sampling date (class `Date`).

Details

The dataset was derived from a GBIF annotated archive (DOI: 10.15468/4qafbu) and processed to extract sampling dates, trap identifiers, and spatial grouping variables.

Only records corresponding to adult females of *Aedes albopictus* were retained. Observations with zero counts were removed, resulting in a zero-truncated dataset suitable for abundance modelling.

The dataset can be used to illustrate lagged associations between environmental variables and mosquito abundance, for example in conjunction with the functions `aggregate_lagged_intervals()`, `fit_models_by_lag()`, and `plotCCM()`.

Source

GBIF occurrence data: doi:10.15468/4qafbu

`fit_models_by_lag` *Fit regression models by lag window on aggregated meteorological predictors*

Description

This function fits a regression model separately for each lag window defined by the `lag_start` and `lag_end` columns of the input data frame. For each lag window, the model is fitted using observations corresponding to different reference dates (`date`), and summary statistics (p-value, betas, sign of effect, R^2, AIC reduction, sample size) are returned for the specified predictor.

Usage

```
fit_models_by_lag(
  data,
  response,
  predictors,
  random = "",
  family = "gaussian()",
  min_n = 10,
  track = F,
  ...
)
```

Arguments

`data`	A data frame containing, at minimum, the columns `lag_start`, `lag_end`, `date`, the response variable, the predictor variable(s) and optional random-effect variables.
`response`	Character string giving the name of the response variable.
`predictors`	Character vector of predictor names. Currently, only a single predictor is supported; providing more than one predictor will result in an error.
`random`	Optional character string specifying random-effects terms to be added to the model formula (without a leading `+`), e.g. `"(1 | site/year)"` or `"(1 | site) + (1 | year)"` (`?glmmTMB::glmmTMB`). If empty (default), a fixed-effect model is fitted.
`family`	Character string. The name of a family function to be used in GLM or GLMM models. Default to "gaussian" (Linear model). see `?stats::family` and `?glmmTMB::family_glmmTMB`
`min_n`	Minimum number of observations required to fit a model. (Currently not enforced; retained for future extensions.)
`track`	If TRUE, lag window is printed in the console before model fitting.
`...`	Additional arguments passed to the underlying modelling function (`glm`, or `glmmTMB::glmmTMB`).

Details

Both fixed-effect and mixed-effect models are supported. The modelling function used depends on the `random` arguments:

- `random = ""`: `glm`
- `random != ""`: `glmmTMB`

For mixed-effects models, marginal R^2 (Nakagawa) is returned. For fixed-effects models, classical R^2 is used.

For each unique combination of `lag_start` and `lag_end`, the function:

1. Subsets the data to the corresponding lag window,
2. Removes rows with missing values in the response or predictor,
3. Fits the specified model,
4. Extracts beta parameter of the linear predictor,
5. Extracts the p-value of the predictor effect,
6. Computes AIC for the specified and null models,
7. Computes the model R^2 (marginal R^2 for mixed models),
8. Records the sign of the estimated effect and the sample size.

The returned table is suitable for lag-window screening, heatmap visualisation, or sensitivity analyses in epidemiological or ecological studies.

Value

A data frame with one row per lag window, containing:

lag_start Start lag index of the aggregation window.

lag_end End lag index of the aggregation window.

predictor Name of the predictor variable.

p_value P-value associated with the predictor effect.

r2 Coefficient of determination (marginal R^2 for mixed models).

betas Estimated beta parameter of the linear predictor.

sign Sign of the estimated predictor effect (-1 or +1).

d_aic AIC reduction compared to the null model.

n Number of observations used to fit the model.

See Also

glmmTMB, r2, r2_nakagawa

---

meteoMPL2023 *Daily meteorological conditions in Montpellier (2023)*

Description

This dataset contains daily meteorological variables for Montpellier (Fréjorgues Airport, WMO station 07643) during the year 2023. The data were derived from 3-hourly SYNOP observations provided by Météo-France and aggregated to daily summaries.

Usage

```
meteoMPL2023
```

Format

A data.frame with one row per day and the following variables:

date Date of observation (class `Date`).

wind_mean Daily mean wind speed ($m \cdot s^{1}$).

wind_min Daily minimum wind speed ($m \cdot s^{1}$).

wind_max Daily maximum wind speed ($m \cdot s^{1}$).

temp_mean Daily mean air temperature (°C).

temp_min Daily minimum air temperature (°C).

temp_max Daily maximum air temperature (°C).

dew.p_mean Daily mean dew point temperature (°C).

dew.p_min Daily minimum dew point temperature (°C).

dew.p_max Daily maximum dew point temperature (°C).

rh_mean Daily mean relative humidity (%).

rh_min Daily minimum relative humidity (%).

rh_max Daily maximum relative humidity (%).

pres_mean Daily mean atmospheric pressure (Pa).

pres_min Daily minimum atmospheric pressure (Pa).

pres_max Daily maximum atmospheric pressure (Pa).

rain_mean Daily mean precipitation (mm).

rain_min Daily minimum precipitation (mm).

rain_max Daily maximum precipitation (mm).

rain_sum Daily total precipitation (mm).

Details

The dataset includes temperature, humidity, wind speed, atmospheric pressure and precipitation, and is intended for use in studies of lagged associations between environmental conditions and ecological or epidemiological time series.

Original 3-hourly SYNOP observations were filtered to retain data from Montpellier–Fréjorgues Airport (WIGOS station 0-20000-0-07643). Air temperature and dew point temperature were converted from Kelvin to degrees Celsius. Negative precipitation values were set to zero prior to aggregation.

Daily statistics (mean, minimum and maximum) were computed for all variables. For precipitation, daily totals are also provided.

This dataset is designed to be used in combination with `aggregate_lagged_intervals()` to generate lagged environmental predictors for ecological or epidemiological modelling.

Source

Météo-France SYNOP data, distributed via data.gouv.fr: https://meteo.data.gouv.fr/datasets/686f8595b351c06a3a790867

`plotCCM` *Plot a cross-correlation map (CCM) from lagged regression results*

Description

This function visualises the strength and direction of associations between a response variable and a lagged predictor across multiple lag windows, using the output of `fit_models_by_lag`. The resulting plot is a two-dimensional "cross-correlation map", where each tile represents a lag window defined by `lag_start` and `lag_end`.

Usage

```
plotCCM(
  data,
  model_outcome = c("r2sign", "r2", "d_aic", "betas"),
  threshold_p = 1
)
```

Arguments

`data`	A data.frame produced by `fit_models_by_lag`, containing at least the columns `lag_start`, `lag_end`, `r2`, `p_value`, and `sign`.
`model_outcome`	Character string specifying the model's outcomes to plot. Either: • `"r2sign"` for the signed coefficient of determination (default), computed as the marginal or classical R^2 multiplied by the sign of the estimated effect, • `"r2"` for the coefficient of determination (R^2), • `"d_aic"` for the delta AIC (compared to the null model) or, • `"betas"` for the estimated beta parameter of the linear predictor.
`threshold_p`	Numeric value giving the p-value threshold above which associations are masked (set to `NA`) in the plot. Default is `1`, meaning that no filtering is applied.

Details

The colour of each tile corresponds to `model_outcome`. Positive associations are shown in red, negative associations in blue, and non-significant or filtered values (using `threshold_p`) are shown in grey.

The lag window yielding the maximum absolute value of `model_outcome` is highlighted with a coloured border.

The x-axis corresponds to `lag_start` (displayed in reverse order), and the y-axis corresponds to `lag_end`. Tiles are coloured using a diverging colour scale centred on zero. Lag windows with `p_value >= threshold_p` are not displayed and appear in grey.

This function does not perform any modelling itself; it is intended solely for visualising results obtained from `fit_models_by_lag`.

Value

A `ggplot2` object representing the cross-correlation map.

See Also

`fit_models_by_lag`, `ggplot`

Index

∗ **datasets**
albopictusMPL2023, 3
meteoMPL2023, 6

aggregate_lagged_intervals, 2
albopictusMPL2023, 3

fit_models_by_lag, 4, *7, 8*

ggplot, *8*
glm, *5*
glmmTMB, *5, 6*

meteoMPL2023, 6

plotCCM, 7

r2, *6*
r2_nakagawa, *6*